Improving Scalability in Parallel File Systems for HEC

HEC FSIO 2
HECURA Research Program
August 7, 2007

Walt Ligon
Clemson University

Project Objectives

- Develop an extensible parallel file system simulation tool
- Study
  - Server-to-server communication
  - Run-time configurable semantics/caching
- Address
  - Scalable metadata
  - Scalable small and unaligned access

Progress To Date

- Focus on development of HECIOS simulation
- Design goals
  - open source
  - object oriented
  - built on existing simulation platform
  - built with existing network infrastructure

Initial Research

- Identified possible tools
  - OpNet
  - NS
  - Omnet++
    - Simulcraft, Inc.
    - Omnest (commercial version)
    - Available since 1995

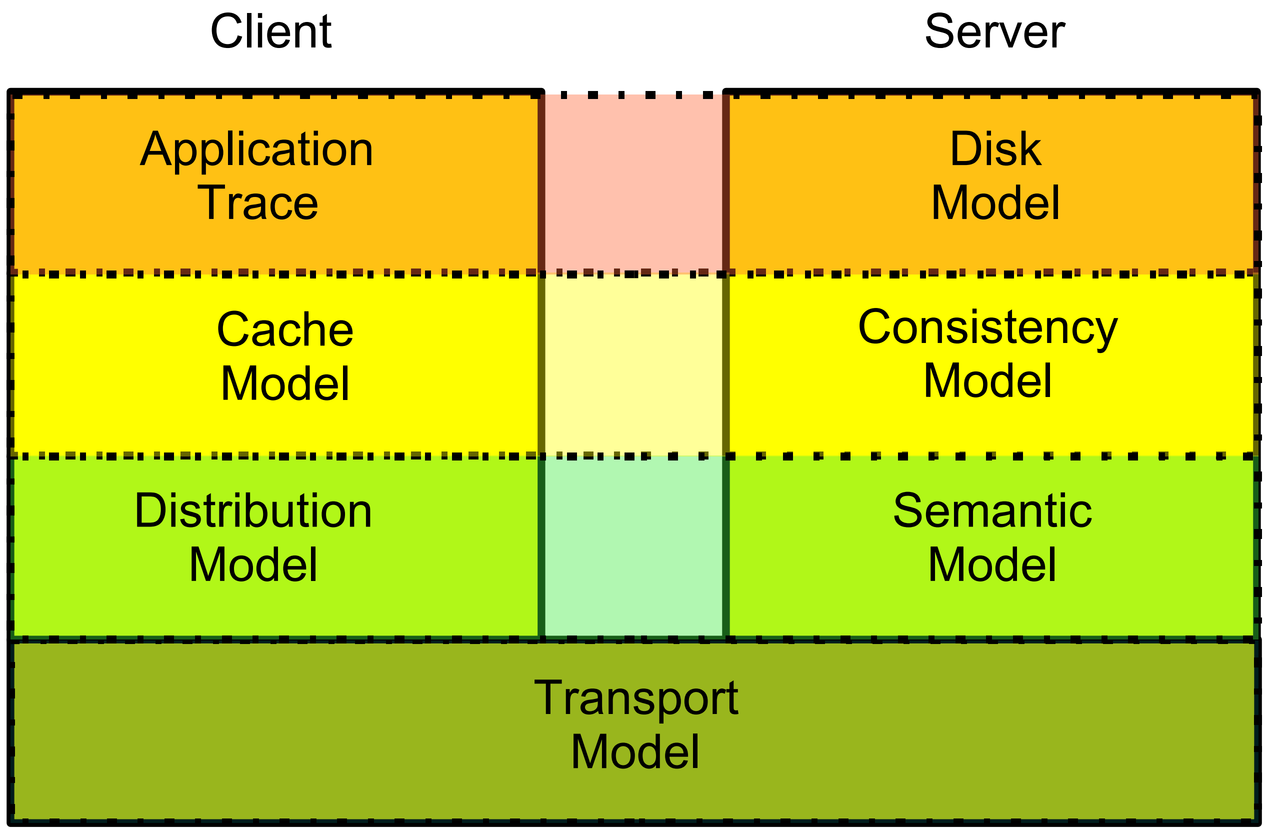
Omnet++

- Written in C++
- Flexible simulation model oriented around communicating modules
- INET - powerful contributed network model
- FsSim - file system/disk contributed model
- Easy to understand

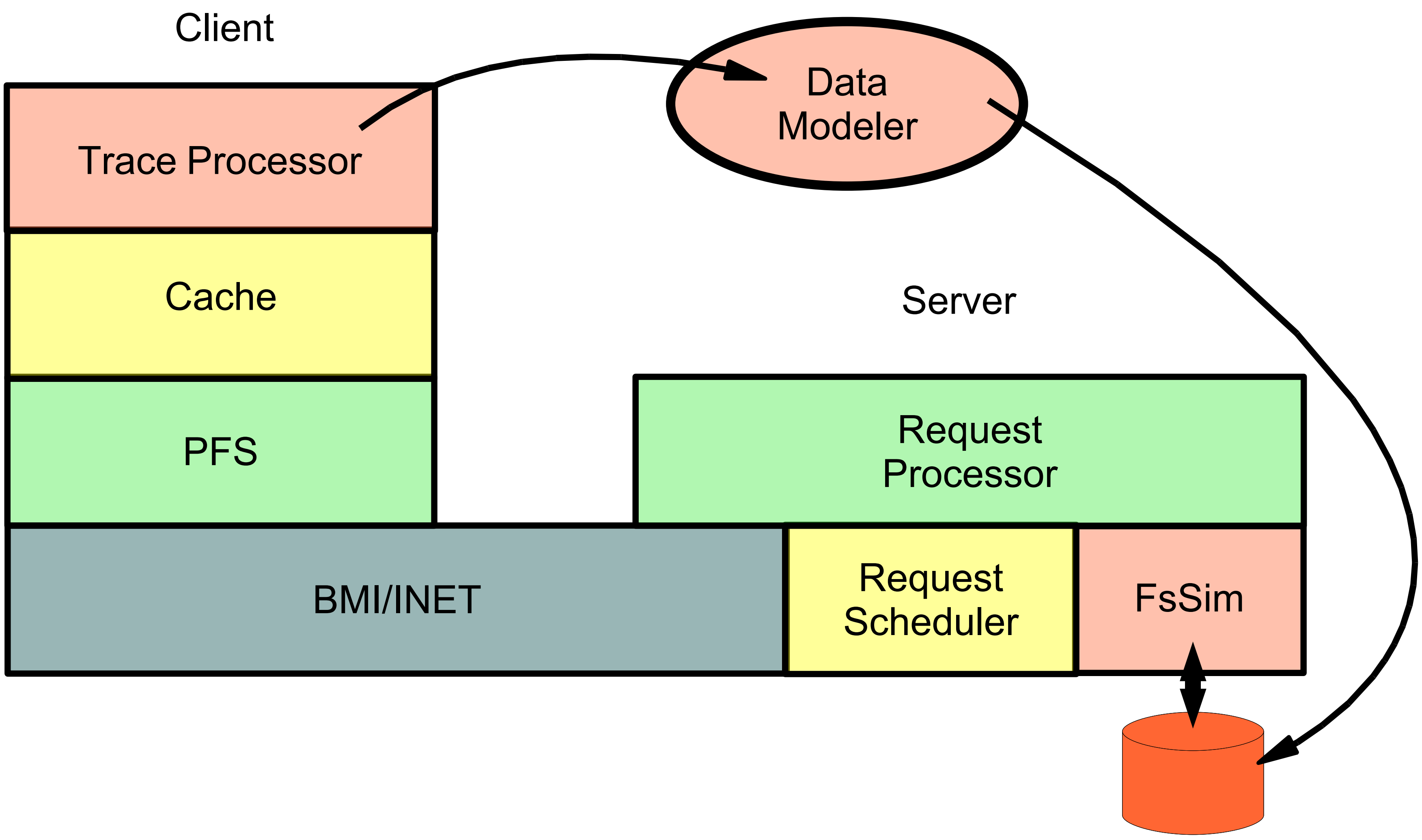
Simulation Model

- Model consists of modules that respond to and send different kinds of messages
- Messages defined using a simple language (supported by a pre-processor)
- FSM abstraction eases programming of message handling routines
- Module configuration using *ned* language (supported by a compiler)
- Other objects easily programmed in C++

SPFS Architecture



SPFS Architecture



Client: Trace Processor

- Reads trace files
  - will have multiple subclasses for different formats
- Produces MPI-IO Request messages
  - roughly based on MPI, represents application level
- Handles most configuration issues

Client: Cache

- Inserts between trace and file system
- Produces and Responds to MPI-IO Requests and Responses
- Responds directly to read and write requests during cache hit
- Updates cache data during reads and writes
- One of our key research areas

Client: PFS

- Responds to MPI-IO Requests
  - Handles all client level file system functions
    - Distribution
    - Flow control
  - Modeled after PVFS request protocol
  - Translates MPI type requests into FS type requests

BMI / INET

- INET is existing TCP/IP model for Omnet++
  - well supported
  - quite accurate
  - easily replaced with other network models
    - someone needs to write those models though
- BMI is interface between modules and network
  - thin interface
  - eases transition to other net models

Server: Request Processor

- Contains bulk of server PFS simulation
  - Large collection of state machines
  - One for each type of request

Server: Request Scheduler

- Key component for handling semantics
  - Checks for available concurrency between requests
  - Initial model prevents concurrent access to file
  - Will study more advanced policies

Server: FsSim

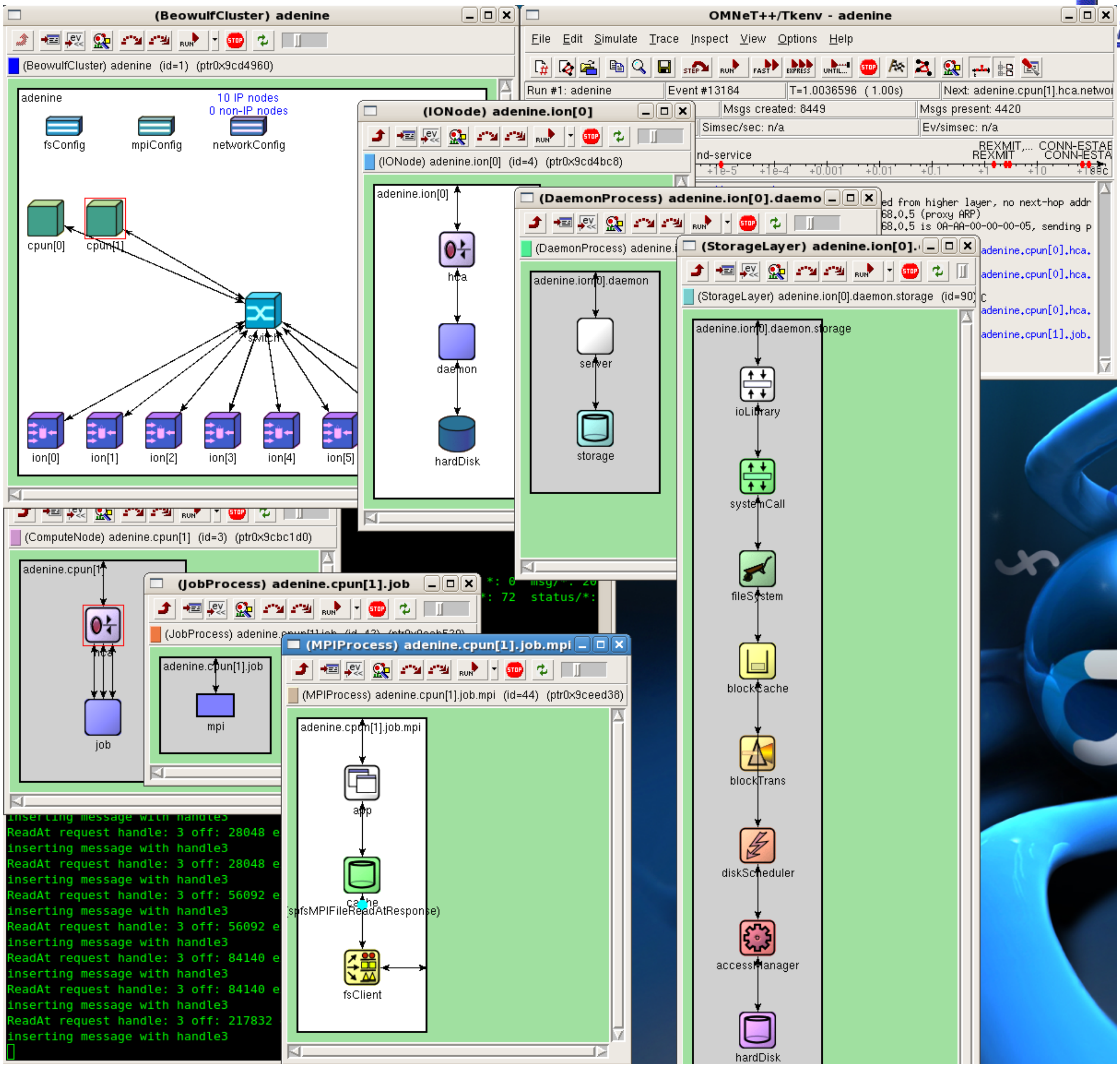
- Existing file and disk subsystem model
- Based on HP disk model from 90's
- Various degrees of accuracy available
- Handles mapping offsets to blocks
- Handles file system cache and track buffer

Data Modeler

- Handles issue of pre-creating files
- Handles pre-locating blocks
- Use probabilistic model for fragmentation
- Pre-processor scans traces to identify files, file sizes, etc.
- Allows PFS parameters (stripe size, etc.) to be specified

Status

- Basic trace processor, cache, PFS implemented and integrated with BMI/INET
- Working request processor and scheduler integrated
- Working FsSim – integrating blocklists
- Data modeler in development


(BeowulfCluster) adenine
(BeowulfCluster) adenine (id=1) (ptr0x9cd4960)
adenine
10 IP nodes
0 non-IP nodes
fsConfig
mpiConfig
networkConfig
cpun[0]
cpun[1]
switch
ion[0]
ion[1]
ion[2]
ion[3]
ion[4]
ion[5]
(IONode) adenine.ion[0]
(IONode) adenine.ion[0] (id=4) (ptr0x9cd4bc8)
adenine.ion[0]
hca
daemon
hardDisk
(DaemonProcess) adenine.ion[0].daemo
adenine.ion[0].daemon
server
storage
(StorageLayer) adenine.ion[0].
(StorageLayer) adenine.ion[0].daemon.storage (id=90)
adenine.ion[0].daemon.storage
ioLibrary
systemCall
fileSystem
blockCache
blockTrans
diskScheduler
accessManager
hardDisk
OMNeT++/Tkenv - adenine
File Edit Simulate Trace Inspect View Options Help
STEP RUN FAST EXPRESS UNTIL... STOP
(ComputeNode) adenine.cpun[1] (id=3) (ptr0x9cbc1d0)
adenine.cpun[1]
hca
job
(JobProcess) adenine.cpun[1].job
adenine.cpun[1].job
mpi
(MPIProcess) adenine.cpun[1].job.mpi
(MPIProcess) adenine.cpun[1].job.mpi (id=44) (ptr0x9ceed38)
adenine.cpun[1].job.mpi
app
cache
(spfsMPIFileReadAtResponse)
fsClient

```
Run #1: adenine    Event #13184    T=1.0036596  ( 1.00s)    Next: adenine.cpun[1].hca.networ
                   Msgs created: 8449          Msgs present: 4420
                   Simsec/sec: n/a             Ev/simsec: n/a
nd-service                                     REXMIT,...  CONN-ESTAB
                                               REXMIT       CONN-ESTA
+1e-5   +1e-4   +0.001   +0.01   +0.1   +1   +10   +100s
```

```
ed from higher layer, no next-hop addr
68.0.5 (proxy ARP)
68.0.5 is 0A-AA-00-00-00-05, sending p
adenine.cpun[0].hca.
adenine.cpun[0].hca.
c
adenine.cpun[0].hca.
adenine.cpun[1].job.
```

```
*: 0  msg/*: 20
*: 72  status/*:
```

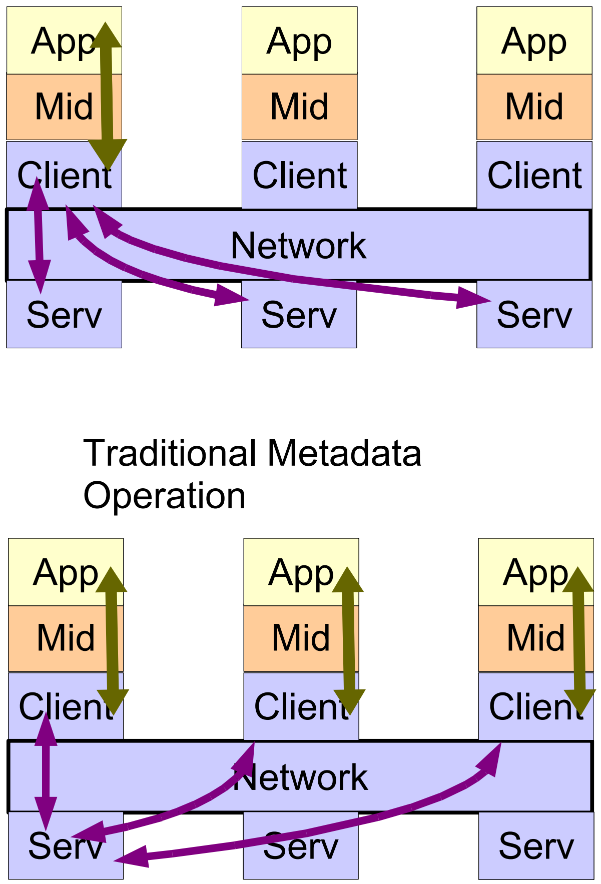
```
inserting message with handle3
ReadAt request handle: 3 off: 28048 e
inserting message with handle3
ReadAt request handle: 3 off: 28048 e
inserting message with handle3
ReadAt request handle: 3 off: 56092 e
inserting message with handle3
ReadAt request handle: 3 off: 56092 e
inserting message with handle3
ReadAt request handle: 3 off: 84140 e
inserting message with handle3
ReadAt request handle: 3 off: 84140 e
inserting message with handle3
ReadAt request handle: 3 off: 217832
inserting message with handle3
```

Concerns

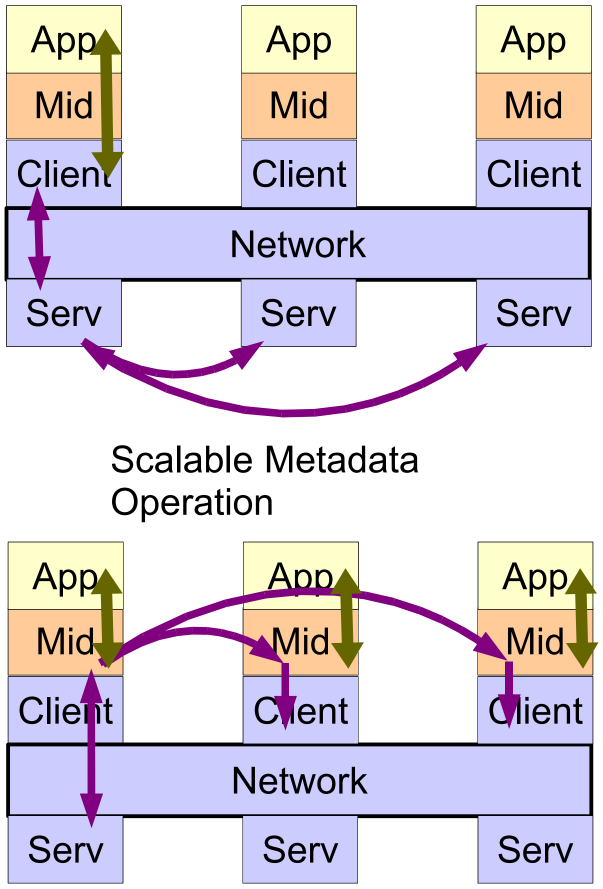
- Traces
  - still unclear what traces will be available or if we will have to produce our own
  - some traces too specialized
  - most traces not parallel
  - UPDATE: HEC researchers promise traces soon!
- Network models
  - really need Myrinet, Infiniband, etc.
  - UPDATE: Infiniband model close to release!

THE END

Scalable Metadata Server-to-Server Communication

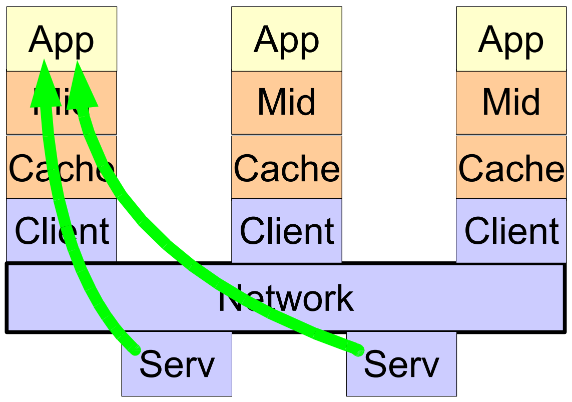


Traditional Metadata Operation

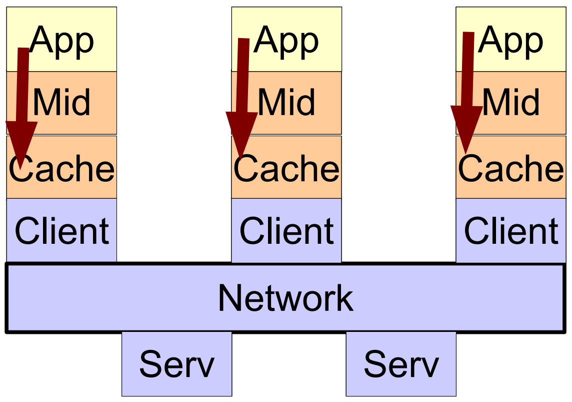


Scalable Metadata Operation

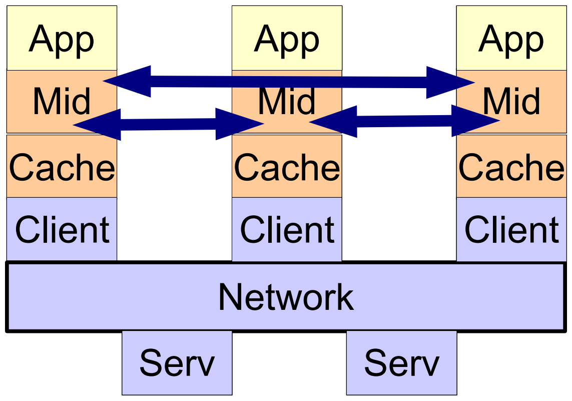
Middleware Managed Cache Weakened Consistency



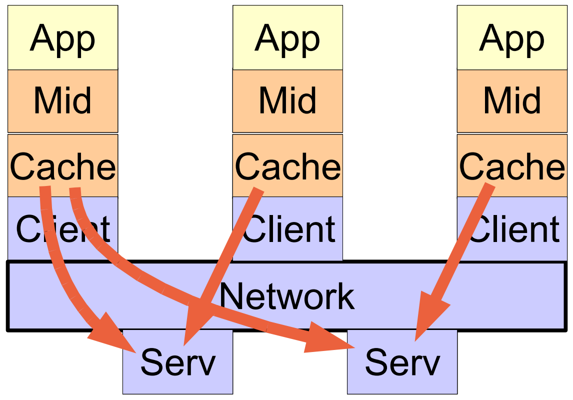
Cold
Read



Writ
e



Synchronization
Event



Write-
back

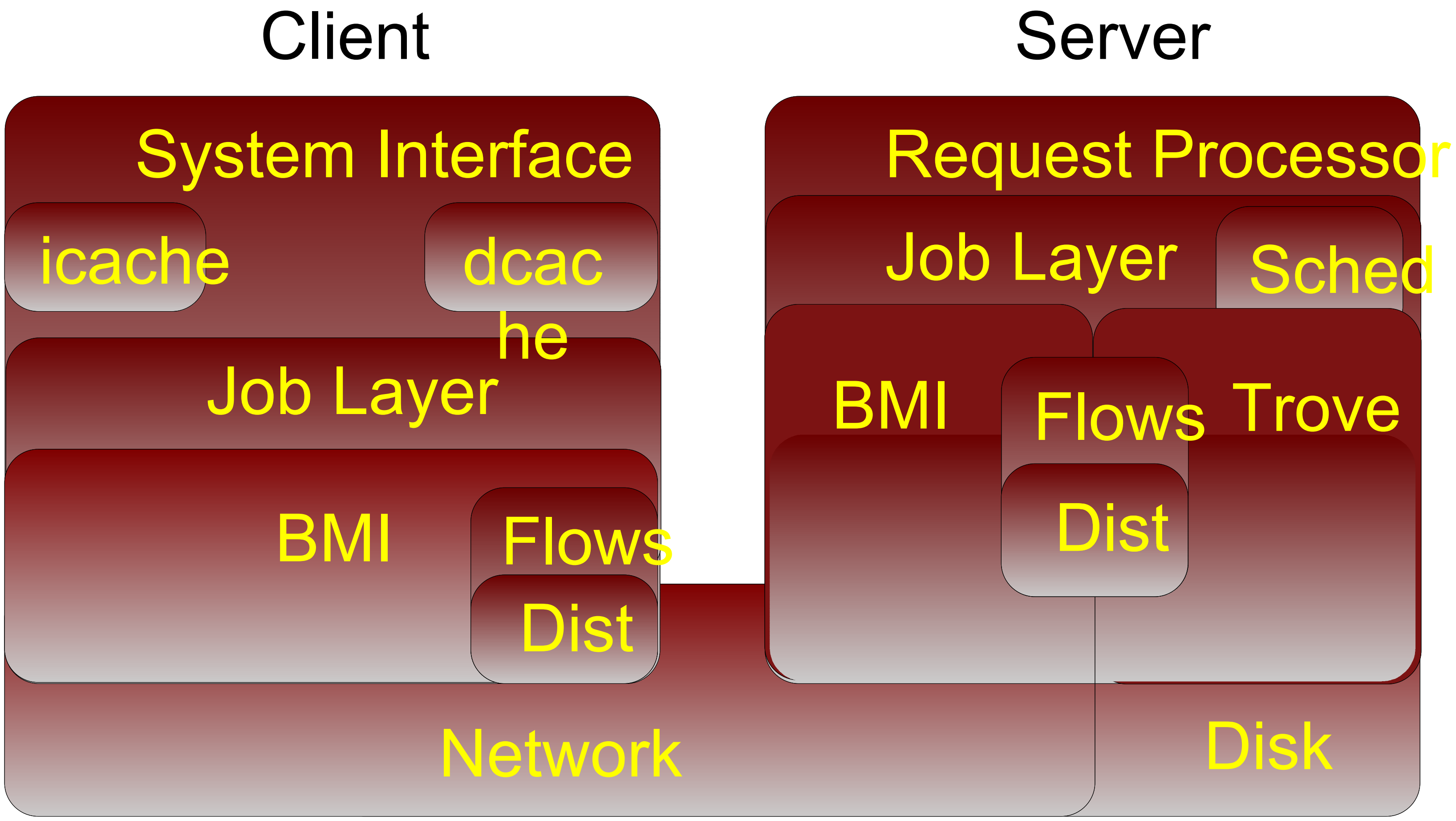
Program Areas Addressed

- scalable metadata operations
- scalable small and unaligned operations
- I/O middleware
- active caching
- server to server communication
- simulation of I/O, file, and storage systems

Research Focus

- Server-to-server communication
  - scalable metadata
- Run-time configurable semantics
  - Lockless SC semantics
  - POSIX versus weakened models
  - caching

PVFS2 Architecture Simulation Model


Client
Server
System Interface
icache
dcache
Job Layer
BMI
Flows
Dist
Network
Request Processor
Job Layer
Sched
BMI
Flows
Dist
Trove
Disk